REMARKS

Claims 1-35 are currently pending in the subject application and are presently under consideration. Claims 6 and 15 have been amended to correct minor informalities, which do not change the scope of such claims. Furthermore, claim 24 has been amended to distinguish the hash from the manifest recited in the respective base claim. Favorable reconsideration of the subject patent application is respectfully requested in view of the amendments and comments herein.

I. Rejection of Claims 1-8, 18, and 23-25 Under 35 U.S.C. § 103(a)

Claims 1-8, 18, and 23-25 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Renaud *et al.*, (US 5,958,051) in view of Buxton (US 6,182,279). Withdrawal of this rejection is respectfully requested for at least the following reasons. Neither Renaud *et al.* nor Buxton alone or in combination teach or suggest applicants' claimed invention.

> To reject claims in an application under §103, an examiner must establish a *prima facie* case of obviousness. A *prima facie* case of obviousness is established by a showing of three basic criteria. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *See* MPEP §706.02(j). The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. *See In re Vaeck,* 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

Independent claims 1, 18, and 23 recite utilization of a "***hash of the contents***" in order ***to verify the integrity of the components employed by application programs during runtime***. The cited references do not teach or suggest such features of applicants' claimed invention. The subject invention facilitates ensuring integrity of assemblies necessary for proper operation of a software application at runtime *via* utilizing ***a hash to ascertaining whether contents of modules*** that make up the assembly have been modified. Thus, for example, an assembly as recited in the subject claims can comprise a dynamic linked library (DLL) containing one or modules, a manifest that contains a list of the modules, and a "***hash of the contents***" of one or more of the modules in the DLL (Note: the hash of the contents is not limited to the modules but can be applied to manifest(s) and other assemblies). The file structure (*e.g.*, providing a manifest with a list of module(s) in an assembly) combined with providing a hash of contents of the module(s) mitigates errors that can occur during instances that a first application task modifies a module that is utilized by a second application task. Furthermore, conventional systems and/or methods are unable to determine existence of such modifications without utilizing the ***hashing of the contents*** (as recited in the subject claims), thereby resulting in inoperability of the second application task and possibly causing costly hardware and/or software damage.

Neither Renaud *et al.* nor Buxton teach or suggest ensuring integrity of assemblies employable by application programs during runtime *via* utilizing a "***hash of the contents***" to verify the contents of at least one module and/or assembly and/or manifest. The Office Action asserts that Renaud *et al.* provides a manifest with "a hash value of data related to at least one module of the list of modules." On the contrary, Renaud *et al.* utilizes a hashing technique known within the art in order to ***hash the file name*** for identification. Renaud *et al.* simply uses one-way hash functions on file names to identify data files to provide security. (*See* col. 7, lines 15-27). Moreover, Renaud *et al.* discloses a conventional file name hashing as a technique known in the art. (*See* col. 7, lines 26-27). The file integrity system described in Renaud *et al.* simply hashes the data file names (not the file contents) in order to verify the integrity. Such technique cannot support verifying integrity of updated data files in which the file names have not been changed (*e.g.*, the contents of DLL file have changed but not the DLL file name) as in applicants' claimed invention. Buxton (which simply relates to storing templates in a component system) does not make up for the aforementioned deficiencies of Renaud *et al.*

Furthermore, Buxton does not teach or suggest ensuring integrity of assemblies employable by application programs during runtime *via* utilizing a "***hash of the*** *contents*" to verify contents of at least one module and/or assembly and/or manifest. On the contrary, Buxton simply provides a component system in which base applications (*e.g.*, components) can be customized by the user in which such customizations (*e.g.*, differences in the base applications) are distributed to another user in the form of templates allowing interaction with the same base applications. (*See* col. 2, lines 18-28). The Office Action states Buxton provides "integrity checking and further discloses the hash of the contents of components making up container." However, Buxton clearly defines a "component" as an OLE control (*See* col. 7, lines 51-54) and a "container" as a stand alone application with embedded OLE controls. (*See* col. 7, lines 65-66). As previously noted, applicants' claimed invention allows integrity of assemblies by providing the assembly with a manifest containing a list of modules in which the manifest is provided with a hash contents of at least one module of the list of modules. Based on the definitions provided by Buxton, there is no assembly containing a manifest with a list of modules. As stated in the specification, an assembly refers to a grouping of files (modules) necessary to perform a particular application, and modules are portion(s) of a computer program that are created to carry out a particular function within the application, and can be utilized alone or combined with other modules in connection with enabling proper operation of the particular application. In other words, Buxton deals with a stand alone application and associated OLE controls. There is no assembly with a manifest containing a list of modules in which the manifest is provided with a hash of the contents of at least one module. Moreover, the Office Action refers to Figures 7-8B in connection to integrity checking. However, Buxton merely provides certification existence based on licensing restrictions. (*See* col. 20, lines 28-41).

It is respectfully submitted that the rationale proffered to combine the references (let alone the deficiencies thereof noted *supra*) is to achieve benefits identified in applicants' specification (employed as a 20/20 hind-sight based roadmap). In essence, the Examiner is basing the rejection on the assertion that it would have been obvious to do something not suggested in the art because so doing would provide advantages stated in applicants' specification. This sort of rationale has been condemned by the CAFC. *See, e.g., Panduit Corp. v. Dennison Manufacturing Co.*, 1 USPQ2d 1593 (Fed. Cir. 1987).

In view of at least the above, it is respectfully submitted that the rejection of independent claims 1, 18, and 23 (and dependent claims 2-8, 24, 25 which respectively depend therefrom) be withdrawn.

II. Rejection of Claims 10-13, and 27-35 Under 35 U.S.C. § 103(a)

Claims 10-13, and 27-35 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Renaud *et al.* in view of Evans *et al.* (US 5,805,899). Withdrawal of the rejection is respectfully requested for at least the following reasons. Neither Renaud *et al.* nor Evans *et al.* alone or in combination teach or suggest applicants' claimed invention.

In particular, Renaud *et al.* and Evans *et al.* do not teach or suggest hashing contents as recited in the subject claims. As discussed above, Renaud *et al.* utilizes a hashing technique known within the art in order to ***hash the file name*** for identification. Renaud *et al.* simply uses one-way hash functions on the file names to identify the data files to provide security as stated *supra*.

Independent claims 10, 27, and 30 (of which claims 11-13, 28-29, 31-35 depend therefrom) recite utilization of a ***hash of the contents*** in order to verify integrity of components employed by application programs during runtime. The subject invention facilitates ensuring integrity of assemblies necessary for proper operation of a software application at runtime *via* utilizing ***a hash to ascertaining whether contents of modules*** that make up the assembly have been modified. The cited references do not teach or suggest such claimed facility of applicants' invention.

Furthermore, Evans does not teach or suggest providing the manifest with a hash of a manifest of at least one referenced assembly of the list of referenced assemblies. As stated in applicants' specification, an assembly refers to a grouping of files (modules) necessary to perform a particular application, and modules are portion(s) of a computer program that are created to carry out a particular function within the application, and can be utilized alone or combined with other modules in connection with enabling proper operation of the particular application. On the contrary, Evans discloses creating a dynamic executable *via* inputting the shared object and relocatable object code into a link-editor. (*See* Fig. 2(b), col. 5, lines 7-13). The dynamic executable has a file format similar to a created shared object, wherein the file comprises a plurality of sections, each section associated with a version dependency section. (*See* Fig. 10, col. 10, lines 32-51). The dynamic executable disclosed in Evans is not equivalent to an assembly as stated in the subject claims. Thus the sections contained in the dynamic executable cannot be equivalent to a list of referenced assemblies that the assembly depends on as recited in applicants' claims. Rather, Evans *et al.* simply discloses utilizing a hash for version names of each version of the versioned object that is contained in the version definition section. (*See* Fig. 6, col. 9, lines 35-37). As the hash is associated only with a version name and no underlying data, ***the hash is not a hash of the contents of a manifest of at least one referenced assembly*** as in applicants' claimed invention.

In view of at least the foregoing, it is respectfully submitted that the cited references do not make obvious applicants' invention as recited in independent claims 10, 27, and 30, and dependent claims 11-13, 28-29, 31-35 which respectively depend therefrom. This rejection should be withdrawn.

III. Rejection of Claims 14-17, and 22 Under 35 U.S.C. § 103(a)

Claims 14-17, and 22 are rejected under 35 U.S.C. §103(a) as being unpatentable over Renaud *et al.* and Evans *et al.* as applied to claims 10 and 16 (for 14-17) and further in view of Buxton. Withdrawal of the rejection is respectfully requested for at least the following reasons. Neither Renaud *et al.*, Evans *et al.* nor Buxton teach or suggest applicants' claimed invention alone or in combination of one another.

Independent claims 10 (from which claims 14-17 depend therefrom) and 22 provide a ***hashing of the contents*** as stated *supra*. Renaud *et al.*, as discussed above, does not teach or suggest ensuring the integrity of assemblies employable by application programs during runtime *via* utilizing a "***hash of the contents***" to verify contents of at least one module and/or assembly and/or manifest. The Examiner relies on Buxton to provide a hash of the contents of modules as stated in independent claim 1. However, Buxton simply provides a component system in which base applications (*e.g.*, components) can be customized by the user in which such customizations (*e.g.*, differences in the base applications) are distributed to another user in the form of templates allowing interaction with the same base applications. (*See* col. 2, lines 18-28). Buxton deals with a stand alone application and associated OLE controls. There is no assembly with a manifest containing a list of modules in which the manifest is provided with a hash of the contents of at least one module. Thus, Buxton does not provide a ***hash of the contents of at least one module*** as recited in the subject claims.

In addition, Evans *et al.* discloses creating a dynamic executable *via* inputting the shared object and relocatable object code into a link-editor. (*See* Fig. 2(b), col. 5, lines 7-13). The dynamic executable has a file format similar to a created shared object, wherein the file comprises a plurality of sections, each section associated with a version dependency section. (*See* Fig. 10, col. 10, lines 32-51). The dynamic executable disclosed in Evans *et al.* is not equivalent to an assembly as stated in the subject claims. Thus, the sections contained in the dynamic executable cannot be equivalent to a list of referenced assemblies that the assembly depends on as recited in the subject claims.

Furthermore, the Office Action states it would be obvious for one skilled in the art to add such hash of contents as taught by Buxton at the end of the assembly to enhance the lay out of the manifest in order to expedite information and facilitate the integrity checking. Again, it is respectfully submitted that such rationale is based on impermissible employment of applicants' specification as a 20/20 hind-sight based roadmap to provide the missing teachings and/or motivation for combination. The cited references do not provide the requisite teachings or motivation therein to establish a *prima facie* case of obviousness.

In view of at least the above, it is readily apparent that the cited references do not make obvious applicants' invention as recited in the subject claims, and this rejection should be withdrawn.

IV. Rejection of Claims 9, 19-21, and 26 Under 35 U.S.C. § 103(a)

Claims 9, 19-21, and 26 are rejected under 35 U.S.C. §103(a) as being unpatentable over Renaud *et al.* and Buxton as applied to claim 1, 18, 23, and further in view of Evans *et al.*, USPN 5,805,899. Withdrawal of the rejection is respectfully requested for at least the following reasons. Neither Renaud *et al.*, Evans *et al.* nor Buxton teach or suggest the applicants' claimed invention alone or in combination of one another.

Independent claims 1, 18 and 23 (from which claims 9, 19-21, and 26 depend therefrom) provide a ***hashing of the contents*** as stated *supra*. Renaud *et al.* does not disclose providing a hashing of the contents. In addition, the Office Action cites Buxton to teach or suggest a signature of components and pointer information to control dependency of components in the aggregate of OLE components to combine at runtime template. However, Buxton rather aggregates a composite object from two or more objects. As discussed earlier, the definitions provided by Buxton do not teach or suggest an assembly containing a manifest with a list of modules.

In regards to the rejection of claim 21, the Office Action states "Official notice is taken that the use of Dynamic Linked Library to effect windows application and user interface functionality was a well-known concept at the time of the invention." Applicants' representative requests a showing of evidence to support such Official Notice. Furthermore, claim 21 is dependent upon claim 18, which provides ***a hash of the contents*** of at least one module. As stated *supra*, the cited references do not teach or suggest providing ***a hash of the contents*** of at least one module as recited in the subject claims.

In view of at least the above, it is respectfully submitted that the rejection of independent claims 1, 18, and 23 and claims 9, 19-21, and 26 which respectively depend therefrom, be withdrawn.

V. Conclusion

The present application is believed to be condition for allowance in view of the above comments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063.

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number listed below.

Respectfully submitted,
AMIN & TUROCY, LLP

Himanshu S. Amin
Reg. No. 40,894

AMIN & TUROCY, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731